**Advanced Scientific Computing Research**
**Applied Mathematics**
**FY 2005 Accomplishment**

# Scalable Solvers – *Toolkit for Advanced Optimization*

Steven Benson*, Jorge Moré, Lois McInnes, and Jason Sarich
Argonne National Laboratory

## Summary

***Discretizations of continuous optimization problems often lead to nonlinear optimization problems with many degrees of freedom. The Toolkit for Advanced Optimization (TAO) enables scientists to apply state-of-the-art optimization algorithms, preconditioned linear equation solvers, automatic differentiation, multilevel methods, and parallel hardware to solve these problems and achieve scalable performance.***

Examples of continuous optimization problems include the variational form of elliptic partial differential equations and obstacle problems from physics and engineering. Obstacle problems, as shown in Figures 1 and 2, minimize the area of a surface fixed at the boundary and stretched over obstacles. Finite element methods approximate these problems by creating a mesh, discretizing the domain, and formulating the optimization problem with a finite number of variables. However, the number of variables required for a sufficient approximation may still be very large and demand computationally intensive methods.

At Argonne we are exploring scalable algorithms for mesh-based optimization problems in the sense that the number of operations required to solve the problem grows linearly with the number of variables: this is an ambitious requirement from a computational viewpoint. Scalable algorithms for mesh-based problems generally required mesh-invariance in the sense that the number of iterations is independent of the granularity of the mesh. In theory, mesh-invariance can be obtained with a traditional Newton's method, but in practice numerous complications arise. The main difficulties are the generation of the mesh, the computation of derivatives, and the preconditioning of the Hessian matrix.

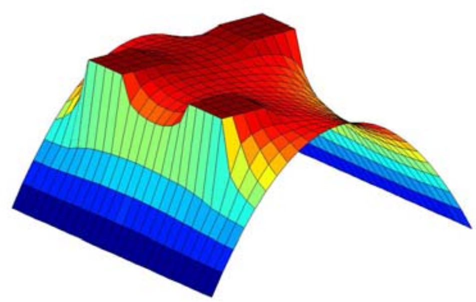

*Figure 1: Obstacle problem on a fine mesh.*

The lack of efficient numerical software makes the development of scalable algorithms in a single-processor environment difficult. Parallel-processor environments magnify these difficulties because the overhead of communication between processors must be balanced with other performance considerations. As a result, many practitioners are left to assemble their own methods. To address

* Mathematics and Computer Science Division, (630)-252-7232, benson@mcs.anl.gov

these problems, the Toolkit for Advanced Optimization (TAO) has implemented efficient optimization algorithms and coupled them with two other software packages developed at Argonne. TAO version 1.8, released in May 2005, uses ADIC to compute derivatives and PETSc to precondition the linear equations. By partitioning the domain over multiple processors, users of TAO can solve mesh-based problems in parallel and achieve scalable performance.

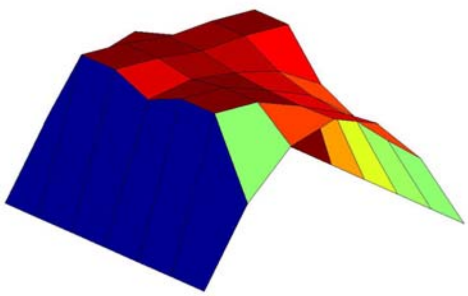

*Figure 2: Obstacle problem on a coarse mesh can be coupled with a fine mesh to provide scalable performance in the optimization solvers.*

Newton's algorithm for the optimization problem requires the first- and second-derivative information from the objective function. Writing code that evaluates the function can be difficult and prone to error. Automatic differentiation is a technique for generating code that computes gradients and higher-order derivatives. Given code that computes the objective function, this software generates additional code that computes the function and its gradient. Using TAO, scientists need only implement the objective function over a single element with a few variables; the toolkit will apply ADIC, the automatic differentiation tool developed at Argonne, to compute the derivatives in parallel.

The performance of optimization solvers depends heavily on the performance of the linear solver in the algorithm. TAO uses PETSc, also developed at Argonne, to apply Krylov methods with a variety of preconditioners such as incomplete LU factorization and additive Schwartz methods.

Mesh sequencing is a technique for solving mesh-based problems that uses the solution of a problem on coarse mesh solution as the initial starting point for a finer mesh. This multilevel method is a standard technique for solving systems of nonlinear differential equations, but with few exceptions, it has not been used to solve mesh-based optimization problems.

**Table 1: Performance of TAO Solver on Obstacle Problem on 16 Processors.**

| Mesh Variables | Iterations | Seconds |
|---|---|---|
| 1121 x 1121 | 4 | 2.5 |
| 2241 x 2241 | 1 | 2.9 |
| 4481 x 4481 | 1 | 9.4 |
| 8961 x 8961 | 1 | 29.4 |

Table 1 shows the time needed to solve the obstacle problem on four distinct meshes. The numbers reflect the use of a Newton solver with an incomplete LU preconditioner, ADIC, and mesh sequencing. The computations used 16 processors from the "Jazz" Linux cluster at Argonne's Laboratory Computing Resource Center. The data shows that as the number of variables increases, the time required to solve the problem increases proportionately. TAO solvers are developed with this kind of scalability in mind.

**For further information on this subject contact:**
Jorge Moré
Argonne National Laboratory
more@mcs.anl.gov
(630)-252-7238